SciVerse ScienceDirect
Disponible en www.sciencedirect.com

Revista Iberoamericana de Automática e Informática industrial 9 (2012) 474–483

www.elsevier.es/RIAI

Framework Heurístico para la Implementación de Sistemas Activos de Reconocimiento de Objetos

E. González[a,*], A. Adán[b], V. Feliú[a]

[a] *Departamento de Ingeniería Eléctrica, Electrónica, Automática y Comunicaciones, Universidad de Castilla La Mancha, E.T.S.I. Industriales, Avda. Camilo José Cela, 13071, Ciudad Real, España.*

[b] *Departamento de Ingeniería Eléctrica, Electrónica y Automática, Universidad de Castilla La Mancha, E.S. de Informática, Paseo de la Universidad 4. 13071, Ciudad Real, España.*

Resumen

Este trabajo presenta un framework para el desarrollo de sistemas activos de reconocimiento de objetos de forma libre. El framework propuesto aborda el problema de incertidumbre presente en los sistemas de reconocimiento de objetos basados en visión monocular mediante un modelo heurístico que permite usar cualquier tipo de vector de características para representar la información de las vistas. De esta manera, se pueden emplear vectores de características que estimen la pose del objeto con mayor precisión que en los tradicionales sistemas estocásticos. La estrategia empleada para el desarrollo del sistema de reconocimiento activo propuesto se basa en agrupar las vistas de los objetos de la base de datos en clusters y, a partir del estudio de la información contenida en ellos, desarrollar de manera eficiente las tareas de clasificación, selección de las posiciones del sensor y el cálculo de la evidencia. El algoritmo de clasificación emplea una máquina de soporte vectorial (SVM) dotando al sistema de reconocimiento de robustez ante pequeñas deformaciones en la apariencia de los objetos por ruido, cambios de iluminación, variaciones en el punto de vista etc. Para la estimación de las posiciones del sensor se utiliza una *D-Sphere* con el objetivo de reducir la incertidumbre empleando el menor número de movimientos. Además, cada cluster es modelado como una *D-Sphere* lo que permite de manera off-line evaluar las diferencias de apariencia, entre objetos ambiguos, según el punto de vista desde el que se les observe . Este método ha sido experimentado en un entorno real con un robot manipulador dotado de una webcam en su efector final.

Palabras Clave:
Reconocimiento de objetos de forma libre, sistema activo de reconocimiento de objetos, SVM, clustering, selección de la próxima vista.

1. Introducción

El uso de sensores de visión acoplados a manipuladores robóticos permite el desarrollo de aplicaciones en entornos industriales tales como ensamblaje, inspección, manipulación de objetos, etc. Sin embargo, lograr aplicaciones flexibles y robustas que trabajen en tiempo real constituye, en estos momentos, el reto de muchos investigadores.

Para el reconocimiento de objetos de forma libre empleando visión monocular (una cámara CMOS o CCD) se necesitan modelos del objeto que contengan una representación de las características del mismo. Existen dos tendencias básicas para obtener dicho modelo (Campbell and Flynn (2001)): las basadas en el modelo estructural (model-based) y las basadas en la apariencia de la imagen (appearance-based). Este artículo se centra en los modelos de representación 3D basados en la apariencia, dado que en términos generales, es posible afirmar que los algoritmos basados en el modelo estructural actualmente son teóricamente capaces de realizar la tarea de clasificar nuevos objetos desconocidos por el sistema, pero en la práctica no permiten su utilización en sistemas visuales de reconocimiento reales debido a la imposibilidad de segmentar de la imagen los datos simbólicos necesarios para su funcionamiento y, por lo tanto, son incapaces de funcionar en un entorno real (Bramao et al. (2011)). Por otro lado, los algoritmos de reconocimiento basados en la apariencia, sí son realizables en sistemas reales siempre que la base de modelos de objetos se haya construido adecuadamente.

Los métodos basados en la apariencia transforman el pro-

*Autor en correspondencia
Correos electrónicos: eliza.glez@gmail.com (E. González), antonio.adan2@uclm.es (A. Adán), vicente.feliu@uclm.es (V. Feliú)
URL: isa.inf-cr.uclm.es/index.php (A. Adán), www.automaticayrobotica.es/ (V. Feliú)

http://dx.doi.org/0.1016/j.riai.2012.09.014

blema de reconocimiento 3D en un problema 2D de reconocimiento de formas ya que el modelo del objeto 3D es asociado a un conjunto de proyecciones adquiridas desde distintos puntos de vista del objeto. La forma del objeto 3D es indirectamente representada por los descriptores 2D de la forma obtenida en cada una de las proyecciones. Entonces el problema de reconocimiento 3D se transforma en un problema de búsqueda de similitudes entre formas 2D. De esta manera el sistema de reconocimiento es independiente de la textura del objeto. Sin embargo, los sistemas de reconocimiento de objetos basados en la apariencia se caracterizan por presentar problemas de incertidumbre debido a la pérdida de profundidad al proyectar la imagen asociada al punto de vista.

La incertidumbre en los sistemas de reconocimiento de objetos se produce durante la interpretación de la vista de un objeto debido a que objetos diferentes pueden tener la misma apariencia según el punto de vista desde el que sean observados (ambigüedad). Otros factores tales como el ruido, variaciones en la iluminación de la escena o errores durante la segmentación incrementan la incertidumbre. En estos casos, lo más aconsejable es obtener información del objeto a reconocer desde otros puntos de vistas de manera tal que la incertidumbre se reduzca. Los sistemas de reconocimiento activo proponen el empleo de estrategias de planificación de los movimientos del sensor para minimizar la incertidumbre empleando el mínimo número de movimientos (Roy et al. (2004)).

Los principales módulos de un sistema activo de reconocimiento son (ver Figura 1):

- **Representación del objeto 3D**. Selecciona el conjunto de puntos de vistas desde los cuales la cámara captura la apariencia del objeto (vistas) para construir la **base de datos**. La información capturada es representada mediante vectores característicos (descriptores de forma) que reducen la dimensionalidad de la imagen.

- **Reconocimiento/clasificación de formas**. Desarrolla el proceso de identificación de la vista (o vistas) hipótesis en la base de datos correspondientes a la vista de la escena.

- **Estrategia activa**. Minimiza los problemas de incertidumbre entre las hipótesis mediante dos procesos: **fusión de la información** y **planificación de los movimientos del sensor**. Debido a que varias hipótesis pueden ser similares a la vista de la escena, el proceso de fusión de la información se encarga de calcular la evidencia de las hipótesis mientras que el proceso planificador de movimientos selecciona la próxima posición del sensor acorde a una función de coste predeterminada.

En el estado de arte sobre sistemas activos de reconocimiento se puede apreciar que la mayoría de los sistemas de reconocimiento activo abordan el problema de la incertidumbre empleando modelos probabilísticos Sipe and Casasent (2002); Borotschnig et al. (1999); Kopp-Borotschnig et al. (2000); Deinzer et al. (2003, 2006). Para aplicar un modelo probabilista es necesario contar con un vector de características (representación

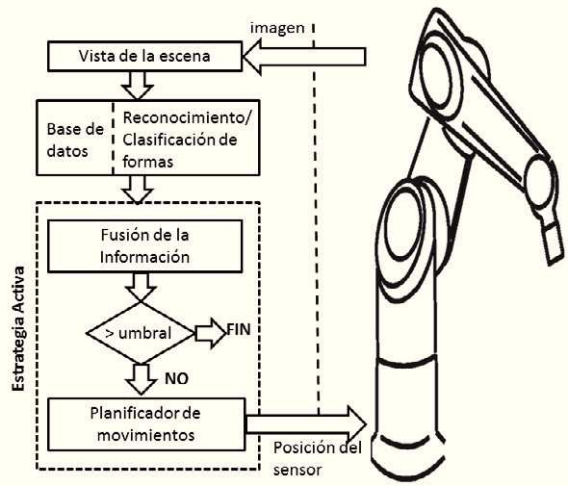


Figura 1: Esquema general de un sistema de reconocimiento activo basado en visión monocular.

de la información de una vista) con propiedades estocásticas, de ahí que la mayoría emplee como vector de características las componentes principales (PCA)(Abdi and Williams (2010)). Sin embargo, en función de la aplicación final del sistema de reconocimiento de objetos, puede que no sea adecuado el uso de PCA. Por ejemplo, en tareas de manipulación de objetos donde se requiere una estimación precisa de la pose del objeto, las PCA no garantizan una precisión aceptable.

Sólo un grupo reducido de trabajos enfoca el desarrollo de sistemas activos de reconocimiento a técnicas no estocásticas Gremban and Ikeuchi (1994); Liu and Tsai (1990); S.A. and A.C (1999); Kovacic et al. (1998). Sin embargo, la eficiencia de estos sistemas es baja cuando la dimensionalidad de la base de datos es elevada o ante variaciones en la escena (entornos no controlados) y errores de segmentación.

La contribución de nuestro trabajo radica en la definición de un framework que permite abordar el problema de ambigüedad para cualquier sistema de reconocimiento de objetos de forma libre basado en el paradigma del reconocimiento activo. Dicho framework aborda el problema de ambigüedad desde un modelo heurístico permitiendo usar cualquier tipo de vector de características para describir la información de las vistas. De esta manera, se pueden emplear vectores de características que permitan estimar la pose del objeto con mayor precisión. La estrategia empleada para el desarrollo del sistema de reconocimiento activo propuesto se basa en agrupar las vistas de los objetos de la base de datos en clusters y a partir del estudio de información contenida en ellos, desarrollar de manera eficiente las tareas de clasificación empleando máquinas vectoriales (SVM), selección de las posiciones del sensor y el cálculo de la evidencia. Además, durante el proceso de experimentación, se desarrolla un análisis comparativo del sistema propuesto con otros sistemas de reconocimiento previamente divulgados en la literatura. Dicho análisis contempla sistemas estócasticos, heurísticos y deterministas.

El artículo está estructurado de la siguiente manera. La Sec-

ción II explica en detalle el modelo de reconocimiento activo propuesto. En la Sección III se discuten los resultados experimentales y finalmente se presentan las conclusiones y los trabajos futuros en la Sección IV.

2. Framework para el reconocimiento activo de objetos

2.1. Introducción

Asumimos que el sistema de reconocimiento de objetos a desarrollar es capaz de reconocer un gran número de objetos de forma libre, existiendo la posibilidad de que algunos de ellos tengan apariencia muy similar entre sí. Las imágenes son capturadas por un sistema de visión monocular capaz de moverse alrededor del objeto. Además, el sistema deberá reconocer los objetos independientemente de la textura de los mismos. Bajo estas condiciones, la ambigüedad/incertidumbre existente entre los objetos a reconocer es elevada.

El estudio de la información contenida entre las vistas de la base de datos permite desarrollar sistemas de reconocimiento más robustos y eficaces computacionalmente. Por ejemplo, las técnicas de clustering (Rai and Singh (2010)) agrupan la información contenida en la base de datos (vistas de la apariencia de los objetos) en grupos con características muy similares. Además, trabajar con una base de datos segmentada en clusters permite optimizar los tiempos de ejecución, ya que la búsqueda se concentra en un sector de la misma (cluster) y no en toda la base de datos.

El proceso de clustering se emplea en este framework con el objetivo de:

1. Estudiar la ambigüedad/incertidumbre de las vistas de la base de datos: A partir de esta información se construye el modelo heurístico que determina la evidencia existente entre las hipótesis correspondientes a una vista de la escena.
   El empleo de un modelo heurístico aporta información al sistema de reconocimiento sobre las características de una vista con respecto a las existentes en la base de datos. La principal ventaja de usar un modelo heurístico en el proceso de reconocimiento es que permite emplear el conocimiento que se obtiene de la base de datos sin necesidad de usar descriptores de forma estocásticos, aportando flexibilidad a la hora de diseñar el sistema de reconocimiento de formas.
2. Estructura del proceso de clasificación: La vista de la escena es clasificada empleando SVM (Máquina de Soporte Vectorial) como perteneciente a una clase. Cada uno de los cluster es considerado una clase.
   Dada la incertidumbre existente en los sistemas de reconocimiento de objetos basados en imágenes 2D, la respuesta del proceso de reconocimiento de formas será un conjunto de vistas con características similares a la vista de la escena. Asumiendo que los objetos son conocidos y que el proceso de clustering agrupa vistas con alto grado de similitud, se puede desarrollar un proceso de aprendizaje (en el caso de SVM consiste en el cálculo de los vectores de soporte SV's) que permita clasificar la vista incógnita en la clase (cluster) donde aparecen vistas similares a ella (hipótesis).
3. Seleccionar la siguiente posición del sensor: El estudio de la ambigüedad entre las vistas de los objetos permite determinar, para un conjunto de objetos hipótesis, cuales son las posiciones del sensor desde las cuales se diferencia su apariencia, o sea, la incertidumbre se minimiza.
   En trabajos previos se abordó el desarrollo de una estructura denominada *D-Sphere* (González et al. (2008b)) cuyo objetivo está enfocado a facilitar la selección de posiciones del sensor que minimizan la incertidumbre entre objetos hipótisis. En este artículo retomamos el uso de la *D-Sphere*. Considerando que el proceso de clasificación selecciona a todos los objetos/vistas que pertenecen a un cluster como hipótesis, el concepto de *D-Sphere* puede aplicarse, off-line, a cada cluster. Es decir, para cada cluster se puede determinar cuál es su *D-Sphere* y de esta forma los costes computaciones de determinar on-line la *D-Sphere* serían reducidos considerablemente. En la sección 2.2.1 explica en detalle este proceso.

El módulo off-line tiene como objetivos principales: 1) Extraer el descriptor de forma de cada vista. 2) Calcular los clusters. 3) Desarrollar el proceso de entrenamiento donde se calculan los vectores de soporte (SV's). 4) Obtener para cada cluster su correspondiente *D-Sphere*.

En el proceso on-line se captura una imagen de la escena y se extraen los descriptores de forma de dicha vista. El proceso de clasificación (SVM) selecciona el cluster cuyos elementos asociados (hipótesis) son los más similares a la apariencia del objeto (vista) presente en la imagen capturada en la escena. Teniendo en cuenta la disimilitud y la información de ambigüedad entre las hipótesis y la vista de la escena, se calcula para cada hipótesis su evidencia. Si ninguna vista de la base de datos satisface el umbral de evidencia pre establecido, el sensor será desplazado hacia una nueva posición. Este proceso se repite hasta encontrar un objeto y una vista del mismo que supere el umbral de confianza.

En la Figura 2 se muestra el esquema general del framework de reconocimiento de objetos propuesto donde se puede apreciar los dos módulos principales.

2.2. Representación de la base de datos

Un método extendido en la literatura para representar el modelo de un objeto 3D mediante su apariencia es el objeto desde un conjunto homogéneo de posiciones que se corresponden con los nodos de una esfera teselada semirregular con origen en el centro de masas (c.d.m) del objeto.

Consideremos un conjunto de N objetos $O = \{o_1, \dots o_N\}$. El modelo sintético de un objeto genérico o_i, $1 \leq i \leq N$ es observado desde distintos puntos de vista correspondientes a los nodos de una esfera teselada con J nodos o sea; desde cada nodo se proyecta una imagen. Las imágenes proyectadas se corresponden con una cámara virtual colocada en cada nodo j ($1 \leq j \leq J$) de una esfera teselada cuya dirección apunta al centro de la esfera donde se encuentra dicho objeto. Sea $I_{i,j}$ la vista obtenida desde el nodo j para el objeto o_i.

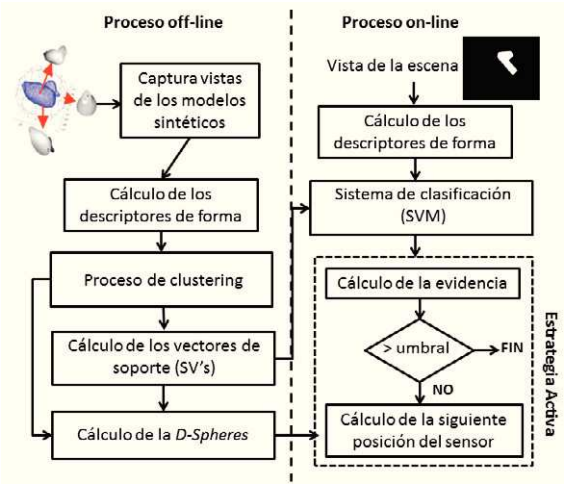


Figura 2: Esquema general del sistema de reconocimiento de objetos propuesto.

La vista $I_{i,j}$ del objeto i es representada mediante un descriptor de formas, ya sea basado en la región o en el contorno, que denotaremos por $\vec{w}_{i,j}$. Este descriptor de forma tiene que cumplir que, al menos, sea invariante a transformaciones geométricas. Ejemplos de descriptores de forma pueden ser las componentes principales (PCA) (Abdi and Williams (2010)), descriptores de Fourier (Zahn and Roskies (1972)), momentos complejos (Flusser (2000)), momentos invariantes (Hu (1962)) etc. (ver (Zitová and Flusser (2003)) para más información).

Supongamos que, como resultado del proceso de clustereing, hemos obtenido un conjunto de R clusters Ω. A cada cluster Ω_r, $1 \leq r \leq R$ han sido asociadas un conjunto de vistas que cumplen que la similitud entre ellas es menor que un umbral predeterminado.

La base de datos quedará definida como $B = \{b_l\}$ $(1 \leq l \leq L$, donde $L = N \times J)$, y cada elemento b_l incluye:

1. El índice del objeto i_l.
2. El índice de la vista j_l.
3. La imagen sintética del objeto i_l tomada desde el nodo j_l: I_l.
4. El descriptor de formas $\vec{w}_l$ de la imagen I_l
5. El índice del cluster r_l al que ha sido asociada la vista I_l
6. Un valor escalar X_l que representa el nivel de ambigüedad. Este nivel de ambigüedad se calcula en base al resultado obtenido con el proceso de clustering.
7. Un valor escalar $\mathcal{E}_l$ que representa la evidencia acumulada .

El valor escalar X_l representa el "nivel"de ambigüedad de una vista y podría interpretarse como una medida de lo diferente o discriminatoria que puede ser una vista con respecto a los demás que aparecen en la base de datos. Definimos X_l como:

$$X_l = \frac{1}{ord(\Omega_{r_l})} \tag{1}$$

donde si la vista b_l pertenece a un cluster Ω con alta densidad, entonces su nivel de ambiguiedad será muy alto. Esta idea se sintetiza en la ecuación (1) donde se proporciona una medida de la ambigüedad de la componente *l-ésima* de la base de datos cuyo cluster asociado es Ω_{r_l}.

Con el objetivo de facilitar la recuperación del índice l asociado a un elemento de la base de datos, se definirá la función $\mu(i, j)$ tal que:

$$l = \mu(i, j) = ((i - 1) \cdot J) + j \tag{2}$$

2.2.1. *Definiendo las* D-Spheres

La representación de un objeto o_i dada su apariencia desde los nodos de una esfera teselada T puede ser interpretada como el mapeo en el *j-ésimo* nodo de la esfera del vector de características (descriptor de formas) de la vista que se observa desde dicho nodo, esto es representado como:

$$T(j) = b_l, l = \mu(i, j) \tag{3}$$

Una *D-Sphere* es una esfera teselada T en la que cada nodo mapea un conjunto de vistas. Dichas vistas se corresponden a un conjunto de objetos que han sido alineados dentro de la esfera con respecto a cierto punto de vista. Si denotamos como Ψ la esfera correspondiente a una *D-Sphere* y j_D los nodos de dicha esfera, entonces

$$\Psi(j_D) = \{b_{l_1}, b_{l_2}...\} \tag{4}$$

Supongamos que queremos alinear dos objetos o_1 y o_2, o sea, hacer corresponder el punto de vista j_2 de o_2 con el punto de vista j_1 de o_1. Entonces, se define la función $A(l_1, l_2, \Upsilon(b_{l_1}, b_{l_2}), j)$ tal que:

$$j_D = A(j_1, j_2, \Upsilon(l_1, l_2), j) \tag{5}$$

calcula para cualquier nodo j del objeto o_2 su correspondiente nodo j_D en la *D-Sphere* después del proceso de alineación. La función $\Upsilon(b_{l_1}, b_{l_2})$ calcula el ángulo de rotación entre la vista j_2 del objeto o_2 y la j_1 del objeto o_1 ($l_1 = \mu(o_1, j_1)$ y $l_2 = \mu(o_2, j_2)$). Nótese que las coordenadas de los nodos de la esfera correspondiente a la *D-Sphere* son los mismos que las coordenadas de la esfera relativa al objeto o_1 y a la de o_2 .

Como resultado de esta alineación al nodo j_D de la *D-Sphere* se han mapeado los elementos b_{l_D} y b_{l_j} :

$$\Psi(j_D) = \{b_{l_D}, b_{l_j}\} \tag{6}$$

donde $l_D = \mu(o_1, j_D)$ y $l_j = \mu(o_2, j)$

Siguiendo este razonamiento, todos los elementos de un cluster son alineados con respecto al primer elemento de dicho cluster. De esta manera, se obtiene para cada cluster una *D-Sphere* Ψ_r (ver en González et al. (2008b) como alinear dos esferas).

Sea el cluster $\Omega_r = \{b_{l_{r,1}}, b_{l_{r,2}}, ..., b_{l_{r,Q}}\}$, donde Q es el número de vistas asociadas al cluster Ω_r. El subíndice r ha sido agregado para resaltar la pertenencia de un elemento a un dicho cluster. La vistas mapeadas al nodo j_D de la *D-Sphere* asociada a este cluster Ψ_r se corresponden con los elementos de la base de datos siguientes:

$$\Psi_r(j_D) = \{b_{\hat{l}_{r,1}}, b_{\hat{l}_{r,2}}, ...b_{\hat{l}_{r,Q}}\} \quad (7)$$

siendo

$$\hat{l}_{r,q} = \mu(i_{l_q}, j_D)$$

y

$$j_D = A(j_{l_{r,1}}, j_{l_{r,q}}, \Upsilon(b_{l_{r,1}}, b_{l_{r,q}}), j), 2 \leq q \leq Q \quad (8)$$

donde $j_{l_{r,1}}$ es la vista correspondiente al primer elemento del cluster r ($b_{l_{r,1}}$) y $j_{l_{r,q}}$ la vista correspondiente al *q-ésimo* elemento del cluster.

2.2.2. *Cálculo de los vectores de soporte vectorial(SV's)*

La máquina de soporte vectorial (SVM por su nombre en inglés *Suport Vector Machine*)(Steinwart and Christmann (2008)) es uno de los mejores métodos de clasificación reportados en la literatura en el apartado de algoritmos de aprendizaje supervisado. Similar a las redes neuronales, SVM incluye aprendizaje por medio del entrenamiento consistente en la optimización de una función de costo. Su ventaja sobre las redes neuronales es que no existe posibilidad de falsos mínimos locales.

Los métodos de SVM buscan un hiperplano que separe de forma óptima a los elementos de una clase de las de otra. Al vector formado por los elementos más cercanos al hiperplano se le denomina vector de soporte vectorial(SV's).

En el caso de conjuntos no linealmente separables (como en la mayoría de los problemas), el empleo de funciones kernel en las máquinas de soporte vectorial ofrece una solución a este problema, mapeando los datos a un espacio de características alto-dimensional, donde se puede hallar más fácilmente un hiperplano de separación.

Durante la fase de entrenamiento son estimados los vectores de soporte (SV's) que separan el conjunto de datos. El framework propuesto considera un clasificador binario para cada cluster. Durante el entrenamiento se han estimado los vectores de soporte que separan un cluster de todos los demás. Por tanto, durante el entrenamiento, los vectores de características de las vistas asociadas a un cluster son etiquetados como clase correcta, mientras que una selección vistas correspondientes a los otros cluster son etiquetadas como clase incorrecta.

Aunque las SVM permiten realizar el proceso de clasificación para múltiples clases, este presenta el inconveniente de que al evaluar una vista y hacerla pasar por varios modelos, la SVM puede categorizar un dato como clase positiva en más de uno de ellos o en ninguno. En este caso, no puede decidirse por ninguna clase y la imagen queda indeterminada. Sin embargo, empleando la estrategia binaria extendida a todos los clusters, este problema es atenuado.

2.3. *Reconocimiento Activo*

El sistema de reconocimiento de objetos por medio de visión activa se lleva a cabo empleando una o varias posiciones del sensor. El criterio de selección de la siguiente posición del sensor (posición y orientación) se basa en la minimización de la ambiguiedad entre las vistas candidatas (hipótesis).

Para desarrollar el modelo de estrategia activa aquí propuesto, es necesario considerar que una esfera imaginaria T_s contiene al objeto de la escena en su centro, y que las posiciones del robot podrán ser los nodos que definen dicha esfera orientadas hacia el centro de ella. El radio de la esfera es conocido y suficiente para que la cámara pueda capturar toda la superficie del objeto sin oclusiones (véase Figura 3)

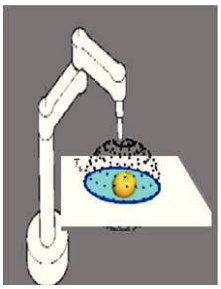


Figura 3: Entorno de trabajo del manipulador. Los nodos de la esfera "imaginaria" alrededor del objeto se corresponden con las posibles posiciones del sensor.

De manera iterativa el sistema activo en cada paso (posición de sensor) obtiene un nuevo conjunto de candidatos. Para estos candidatos se calcula su evidencia y, en caso necesario, se estima la próxima posición del sensor. Dicha posición y orientación corresponden a alguno de los nodos de una esfera teselada predefinida T_s.

La inicialización de este proceso ($k = 0$) desarrolla las siguientes operaciones:

1. La evidencia acumulada $\mathcal{E}_l$ se inicializa a cero para todos los elementos de base de datos $b_l, 1 \leq l \leq L$.
2. Desplazar el sensor hasta obtener una vista centrada del objeto en la escena (véase (Kragic and Christensen (2002)) para más información).
3. Desplazar la esfera de trabajo hasta la posición en que se encuentra el robot.

El proceso iterativo para el reconocimiento activo se lleva a cabo de la siguiente manera. Supongamos que hemos completado el paso de la iteración $k-1$. Como resultado de este paso, hemos obtenido: 1) la base de datos hipótesis $B^{(k)}$, con la evidencia acumulada actualizada (la evidencia acumulada en la hipótesis de l después de $k-1$ pasos se denota como $\mathcal{E}_l^{(k-1)}$). El algoritmo comienza desde $k = 1$, y realiza el paso k a partir del paso $k-1$ por llevar a cabo la siguientes dos tareas de manera secuencial. La posición del sensor en el paso k se corresponde con el nodo $j^{(k)}$ de la esfera T_s.

2.3.1. *Tarea 1: Obtención de las vistas hipótesis y fusión de la información*

Las acciones involucradas en esta tarea son los siguientes:

1. Capturar una imagen $I^{(k)}$
2. Extraer el vector de características $\vec{w}^{(k)}$
3. Clasificar $\vec{w}^{(k)}$. El proceso de clasificación evalúa para cada cluster, si $\vec{w}^{(k)}$ pertenece a ese conjunto. Por tanto, el resultado del proceso de clasificación es el índice del cluster ganador r^*. Las vistas hipótesis son el subconjunto $B^{(k)}$ determinado por:

$$B^{(k)} = \{b_l\}, \forall b_l \in \Omega_{r^*}$$

4. Calcular la evidencia de las vista pertenecientes a $B^{(k)}$

$$\mathcal{E}_l^{(k)} = \mathcal{E}_l^{(k-1)} + [(1 - D(\vec{w}_l, \vec{w}^{(k)})) \cdot X_l] \quad \forall b_l \in B^{(k)} \quad (9)$$

donde $D(\vec{w}_l, \vec{w}^{(k)})$ es una medida de similutud. Ejemplos de este tipo de medidas son: distancia de Manhattan, Coseno del ángulo, la norma L_2, etc. (Janoos (2006)).

5. Comprobar si alguna vista candidata satisface:

$$\mathcal{E}_l^{(k)} \geq \eta, \forall l \in B^{(k)} \quad (10)$$

Si la ecuación (10) se cumple, el objeto/vista correspondiente a la vista l^* se considera como solución al problema de reconocimiento. En caso contrario, la tarea 2 es llevada a cabo.

2.3.2. Tarea 2: Planificación de los movimientos del sensor

En condiciones ideales, la planificación de los movimientos del robot considera solamente aquellas posiciones donde se reduce la incertidumbre entre las hipótesis. Sin embargo, en aplicaciones prácticas, otros dos parámetros deben ser considerados: espacio de trabajo del robot y el coste del movimiento.

El parámetro coste de movimiento g_j mide el coste requerido por el robot para trasladar el sensor de una posición a otra. Este parámetro es definido por la función $\mathcal{G}(j^{(k)}, j)$

$$g_j = \mathcal{G}(j^{(k)}, j) \quad (11)$$

Esta función se puede expresar en base a los consumos de energía del robot, al tiempo necesario para mover el robot de una posición a otra o la distancia entre dos nodos.

El espacio de trabajo del robot ς_j es una función binaria que expresa si un nodo de la esfera T_s puede ser alcanzado o no por el robot.

Asumiendo que el espacio de trabajo de robot no varía dinámicamente durante el proceso de reconocimiento, g_j y ς_j pueden ser calculados off-line.

El grado de incertidumbre β_j^r entre un conjunto de vistas que se observan desde cierta posición, también puede ser determinado off-line empleando el modelo de la *D-Sphere*. Dado que cada nodo de Ψ_r mapea un conjunto de vistas, se puede calcular el grado de incertidumbe existente entre las vistas mapeadas en cada nodo como:

$$\beta_j^r = minX_l, \forall b_l \in \Psi_r(j) \quad (12)$$

Una vez calculados β_j^r, g_j, ς_j, el proceso de selección de la próxima vista se desarrolla llevando a cabo la evaluación de una función de coste. La siguiente posición del sensor se corresponde con el nodo j_o que maximiza la función de coste $\mathcal{J}(\rho_j, g_j, \varsigma_j)$

$$\mathcal{J}(\rho_j, g_m, \varsigma_j) = \varsigma_j \cdot \mathcal{J}'(\rho_j, g_j) \quad (13)$$

donde ρ_j es la incertidumbre esperada si el sensor se coloca en la posición j dado el conjunto de hipótesis obtenidas. Por tanto, este valor se corresponde con el β_j^r una vez que la *D-Sphere* Ψ_{r^*} es alineada con la esfera de la escena ($\bar{\Psi}_{r^*}$).

$$\bar{\Psi}_{r^*}(j_D) = \Psi_{r^*}(j) \quad (14)$$

donde $j_D = A(j^{(k)}, j_{l_{r^*,1}}, \Upsilon(b^{(k)}, b_{l_{r^*,1}}), j)$ y $l_{r^*,1}$ es el nodo correspondiente a la primera vista del cluster r^*. Recuerdese que todas las otras vistas del cluster fueron alineadas a dicho nodo. De esta alineación se obtiene que:

$$\rho_j = \beta_{j_D}^{r^*} \quad (15)$$

Una vez que se ha determinado la nueva posición del robot j_o, se realiza la actualización de la evidencia donde las vistas esperadas en la nueva posición del sensor "heredan" la evidencia de las hipótesis en las posiciones anteriores del sensor. Para llevar el historial de todas los objetos/vistas hipótesis, se utiliza la *D-Sphere* Γ. En la posición k del sensor, $\Gamma^{(k)}$ contiene el historial de todas objetos hipótesis alinenados al objeto de la escena. Entonces,

$$\Gamma^{(k)} = \Gamma^{(k-1)} \cup \bar{\Psi} \quad (16)$$

La evidencia acumulada es actualizada para las vistas mapeadas en $\Gamma^{(k)}$ al nodo j_o. Primeramente, actualizamos la evidencia acumulada con el valor de la evidencia calculada para cada vista:

$$\mathcal{E}_{\bar{l}_v}^{(k)} = \mathcal{E}_{l_v}^{(k)}, \forall v, 1 \leq v \leq V \quad (17)$$

siendo V el número de hipótesis mapeadas en $\Gamma^{(k)}$, $\bar{l}_v$ es la $v-$ésima vista mapeada en el nodo j_o de $\Gamma^{(k)}(j_o)$ y l_v es la $v-$ésima vista mapeada sobre el nodo $j^{(k)}$ de $\Gamma^{(k)}(j^{(k)})$.

La actualización de la evidencia empleando el historial de todos los movimientos del robot garantiza que, si en alguna posición la hipótesis estimadas son erróneas, no se pierda la información del proceso desarrollado en las posiciones anteriores.

La Figura 4 ilustra con un ejemplo el comportamiento del sistema propuesto en caso de incertidumbre por la ambigüedad. Para la primera vista capturada de la escena ($k = 1$), el sistema identifica un cluster en el cual se encuentran vistas relativas a dos objetos diferentes. Para cada candidato se calcula la evidencia ($\mathcal{E}^{(k)}$). La función de decisión es evaluada y la siguiente posición del sensor es determinada. Una vez determinada la siguiente posición, la evidencia acumulada es actualizada para las vistas esperadas en la nueva posición del sensor. El sensor es desplazado hacia una nueva posición ($k = 2$) y una nueva imagen de la escena es capturada. Una vez más, dicha vista es clasificada en un cluster y el valor de la evidencia es estimada para cada uno de los miembros de dicho cluster. Finalmente, la evidencia acumulada para el candidato ($o_{1,16}$) satisface el umbral (η) predefinido y el proceso de reconocimiento se detiene.

3. Experimentación

La implementación del framework de reconocimiento propuesto se desarrolló empleando 18 objetos de forma libre de la base de datos 3DSL (UCLM (2011)). Para cada uno de estos objetos se obtuvo su correspondiente modelo sintético empleando un escáner láser 3D VI-910 de Konica Minolta. La Figura 5(a) muestra los modelos sintéticos de los objetos empleados en la base de datos.

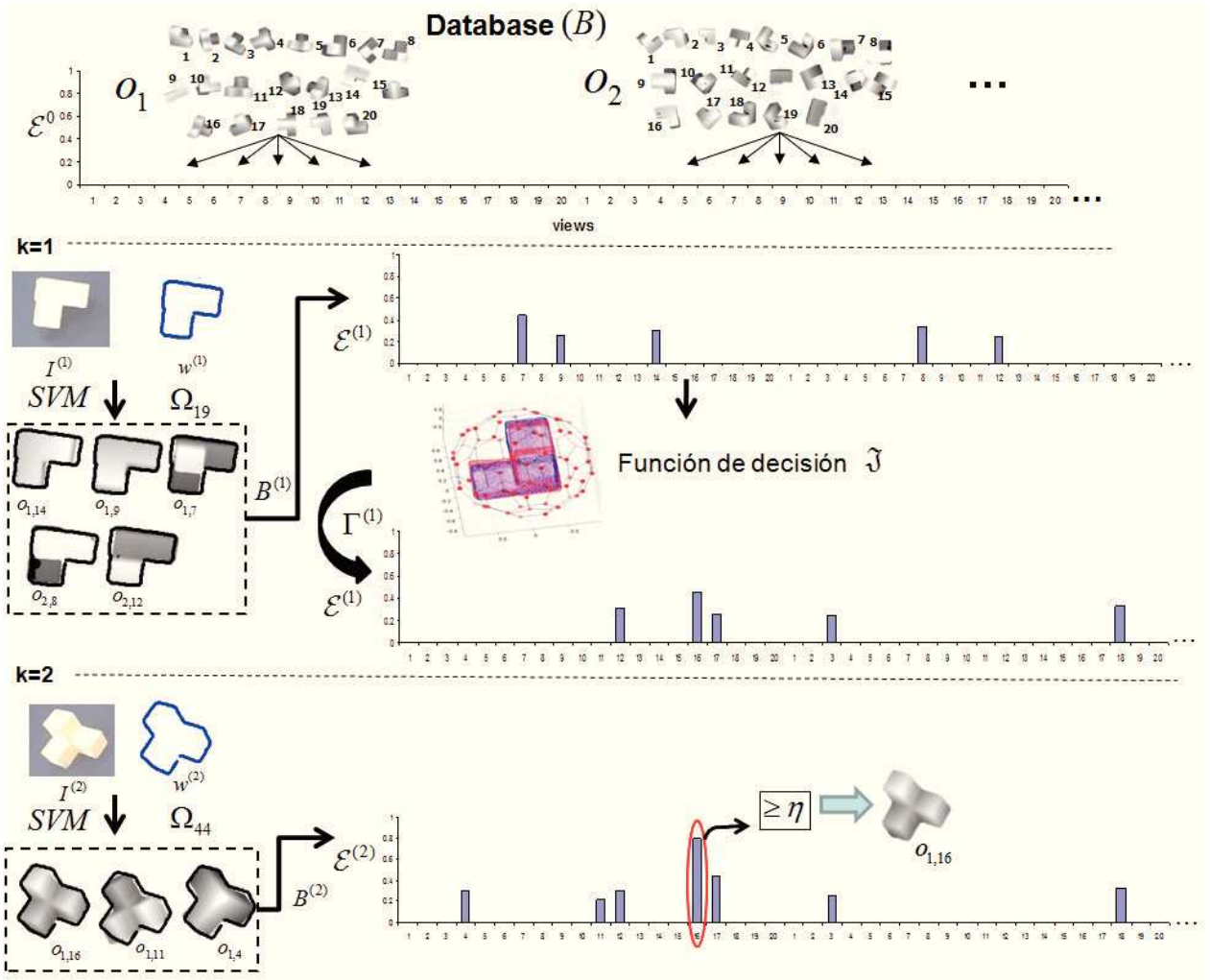


Figura 4: Ejemplo del desempeño del framework propuesto en caso de incertidumbre por la ambigüedad.

Los objetos han sido modelados capturando su apariencia desde distintos puntos de vista utilizando las proyecciones de dichos objetos sintéticos desde una esfera teselada con 80 nodos. Tanto la selección del número de nodos en la esfera como la implementación del módulo relativo al sistema de reconocimiento de formas está basado en los resultados obtenidos en González et al. (2012). La representación de formas se basa en el empleo de descriptores de Fourier (Zahn and Roskies (1972)) empleando 128 harmónicos (ver en González et al. (2008a) como representar una silueta mediante descriptores de Fourier y el cálculo de su pose).

En la Tabla 1 se muestran las funciones empleadas para implementar el framework.

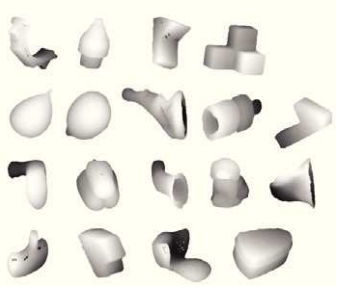
En el proceso de clustering se emplearon los 16 harmónicos más importantes del vector de descriptores obteniéndose en total 73 clusters. En González et al. (2004) se demuestra cómo un pequeño subconjunto de harmónicos puede representar las características más importantes de una silueta. La técnica de clustering implementada se basa en el algoritmo de agrupamiento jerárquico desarrollado por Yuan and Street (2007). A diferencia del tradicional método k-means (Kanungo et al. (2002)) donde el número de clusters es fijado, en los métodos de clustering jerárquicos el agrupamiento de siluetas se lleva a cabo midiendo, mediante la distancia Euclídea, la similitud entre las siluetas.

En la fase de entrenamiento se ha tratado de hacer al sistema robusto ante entornos no controlados, simulando posibles deformaciones en la vistas. Para ello, se añadieron a las imágenes de entrenamiento diferentes clases de ruido (gaussiano, sal y pimienta) y filtros para alterar la forma. Además se capturaron imágenes desde puntos de vista con coordenadas próximas a las de cada nodo de la esfera teselada, pero manteniendo la orientación del sensor hacia el centro del objeto, para simular deformaciones dadas por variaciones de la posición del sensor. En total, se usaron 8 siluetas de entrenamiento para cada elemento de un cluster. La Figura 5(b) muestra ejemplos de imágenes usadas durante el entrenamiento. Para el cálculo de los vectores de soportes se empleó la función de base radial Gaussiana

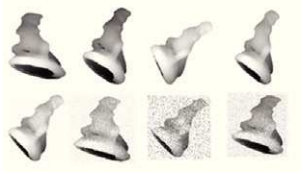
Tabla 1: Métodos usados en la implementación del framework

Función	Método empleado
$\vec{w}$	$\vec{w} = FD$, 128 harmónicos (González et al. (2008a))
$D(v_l, v^{(k)})$	$\sqrt{\Sigma(w_l, w^{(k)})^2}$
$\Upsilon(w_l, w^{(k)})$	ver (González et al. (2008a))
$g_m = \mathcal{G}(j^{(k)}, j)$	$\sqrt{\Sigma(\varpi^{(k)}, \varpi_j))^2}$ donde $\varpi^{(k)}$ es las coordenadas cartesina de la posición actual del robot y ϖ_j la coordenada cartesiana del nodo j
ς_j	Configurado off-line considerando la accesibilidad de sensor al nodo j
$\mathcal{J}$	$\varsigma_j \cdot max_j(\rho_j \cdot g_j)$

como kernel(Cristianini and Shawe-Taylor (2010)) con un error mínimo de 0.030.



(a)



(b)

Figura 5: a) Ejemplos de modelos sintéticos. b) Ejemplos de imágenes empleadas durante proceso de obtención de los SV's.

Las pruebas desarrolladas en esta sección se centran en la cuantificación del sistema de reconocimiento activo propuesto mediante la evaluación de varios parámetros tales como: tasa de reconocimiento (R), eficiencia computacional (C), y el error de estimación de la pose (P_o). Los dos primeros parámetros pueden ser calculados mediante el uso un escenario real de trabajo (plataforma robótica), pero el error de estimación de la pose necesita ser evaluado por medio de un simulador ya que no es posible determinar en una escena real la pose de un objeto con respecto a su correspondiente en la base de datos y medir con exactitud dicho valor.

Varios sistemas de reconocimiento activo han sido implementados con el objetivo de comparar el rendimiento de nuestro sistema con respecto a otros ya publicados. Esta selección se llevó a cabo considerando: tipo de estrategia activa (determinista o probabilista) y repetitividad de los resultados publicados. En total cinco sistemas fueron implementados identificados como AR1, AR2,...,AR5. AR1 se corresponde con el sistema activo propuesto. AR2 es la técnica publicada por Borosting et ál. (Borotschnig et al. (1999)) basado en un modelo probabilístico (Bayes) que emplea en este caso descriptores de forma de tipo PCA (Abdi and Williams (2010)). AR3 es el sistema desarrollado por Hutchinson y Kak (S.A. and A.C (1999)) el cual es estocástico también (Dempster Shafer), pero los objetos son modelados por un gráfo de aspecto. AR4 corresponde a una implementación del framework propuesto por Kovacik et ál. (Kovacic et al. (1998)) en el que la estrategia activa es determinista y hemos empleado también descriptores de Fourier para la representación de formas. AR5 se corresponde con un sistema activo similar al presentado en este artículo (González et al. (2008a)).

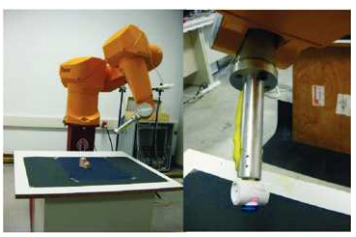
Tabla 2: Principales modelos empleados en cada uno de los sistemas de reconocimiento activo a comparar

Sistema	Descriptor	Estrategia activa
(AR1)	FD	Heuristico
(AR2)	PCA	Probabilista
(AR3)	PCA	Probabilista
(AR4)	FD	Determinista
(AR5)	FD	Heurístico

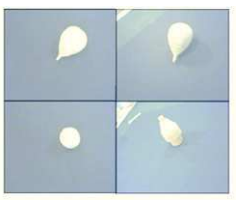
A continuación se desarrolla un análisis comparativo de los sistemas de reconocimiento implementados tanto en una plataforma robótica como empleando imágenes simuladas.

3.1. Plataforma Robótica

Las pruebas de validación del sistema propuesto para el reconocimiento de objetos de forma libre se han realizado utilizando un Robot Stäubli RX 90 con una micro cámara Jai-CVM1000 en el extremo de su brazo (ver Figura 6(a)). Este sistema es capaz de controlar la posición y orientación de la cámara. En la Figura 6(b) se pueden apreciar varios ejemplos de imágenes capturadas por el robot.



(a)



(b)

Figura 6: a) Estructura del experimento. b) Ejemplos de imágenes capturadas durante las pruebas de reconocimiento activo.

La Tabla 3 muestra la comparación entre los distintos sistemas de reconocimiento (AR1, AR2, AR3, AR4, AR5) con respecto a la tasa de reconocimiento (R) y la eficiencia computacional (C). En este caso, el coste computacional (C) se calcula como $C = W \cdot T$, donde W es el número medio de posiciones del

sensor y T es el costo computacional en cada posición del sensor. Los resultados obtenidos se corresponden a la realización de 300 pruebas.

Tabla 3: Comparación entre diferentes sistemas de reconocimiento activo en la plataforma robótica

Sistema	T (s)	W (%)	C(s)	R (%)
(AR1)	0.18	3.8	0.68	95
(AR2)	0.21	4.6	0.86	96
(AR3)	0.19	4.3	0.82	95
(AR4)	0.17	5.2	0.88	89
(AR5)	0.20	5.5	1.10	92

Comparando los resultados obtenidos en la Tabla 3 podemos concluir que AR1 emplea un número menor de movimientos del sensor que AR2 y AR3, por lo que su coste computacional es menor. Con respecto a AR4 y AR5, es destacable la superioridad de AR1 en cuanto a la tasa de reconocimiento. En general, AR1 presenta la mejor relación entre coste computacional y tasa de reconocimiento.

3.1.1. Pose evaluación empleando objetos sintéticos

Los experimentos desarrollados en la plataforma del robot no permiten evaluar el error de estimación de la pose de objeto de manera precisa. Empleando objetos sintéticos, en vez de los reales de la escena, es posible calcular el error de estimación. Con el fin de evaluar el comportamiento de AR1, AR2, AR3, AR4 y AR5, hemos llevado a cabo una serie de pruebas utilizando imágenes simuladas pertenecientes a las vistas proyectadas de los objetos sintéticos de la base de datos una vez que han sido rotados.

Sea $\hat{x}_i, \hat{y}_i$ y $\hat{z}_i$ las coordenadas asociados con la nube de puntos del objeto ($\hat{o}_i$) y $\hat{\hat{x}}_i, \hat{\hat{y}}_i$ y $\hat{\hat{z}}_i$ las coordenadas de los candidatos después de aplicar una transformación $\hat{T}$ estimada por el sistema de reconocimiento de activo. El error durante la estimación de la pose se calcula por la expresión:

$$P_o = \frac{\sqrt{\sum(\hat{x}_i - \hat{\hat{x}}_i)^2 + (\hat{y}_i - \hat{\hat{y}}_i)^2 + (\hat{z}_i - \hat{\hat{z}}_i)^2}}{\hat{n}} \qquad (18)$$

donde $\hat{n}$ es el número de puntos en la muestra.

Tabla 4 muestra la eficiencia computacional (C), la tasa de reconocimiento (R) y la estimación de la pose (P_o) resultados en el simulador para 54 objetos rotados al azar.

Tabla 4: Comparación entre diferentes sistemas de reconocimiento activo empleando objetos sintéticos

Active model	C(s)	R (%)	$P_o(cm)$
(AR1)	0.47	99	0.012
(AR2)	0.73	99	0.244
(AR3)	0.70	98	0.285
(AR4)	0.42	93	0.030
(AR5)	0.89	96	0.029

Comparando los resultados obtenidos en Tabla 4 con los mostrados en Tabla 3, se puede apreciar la robustez de cada uno de estos sistemas ante ruido y pequeñas deformaciones de la apariencia de un objeto en la escena. Sistemas como AR4 y AR5 son más sensibles que AR2 y AR3. En el caso de AR1, el empleo de vistas deformadas durante el entrenamiento permite que la clasificación se desarrolle de manera más robusta.

En cuanto a la estimación de la pose, se puede apreciar en la Tabla 4 que los sistemas basados en descriptores de Fourier (AR1, AR4 y AR5) presentan mejores resultados que AR2 y AR3.

3.2. Análisis de los resultados

Los resultados experimentales demuestran que, para sistemas basados en modelos no estocásticos, AR1 presenta un mejor comportamiento y, en el caso de los modelos estocásticos, se consigue una tasa de reconocimiento muy similar pero el costo computacional y la precisión del cálculo de la pose de un objeto es mucho mejor en nuestro sistema que en los sistemas estocásticos. La razón está dada principalmente porque el cálculo de los parámetros de pose (rotación, traslación y escalado) entre dos formas representadas por PCA tiene baja precisión. Para conseguir una mejor estimación de la pose en un sistema estocástico habría que aumentar considerablemente el número de vistas de la base de datos lo que incrementaría el coste computacional del sistema.

Profundizando el análisis comparativo entre nuestro sistema con respecto a AR4, AR5 (basados en un modelos no estocásticos), AR1 es mucho más robusto en cuanto a variaciones en la escena. El empleo de la máquina de soporte vectorial permite identificar satisfactoriamente, en la mayoría de los casos, el cluster donde se encuentran las hipótesis correctas. En el caso de AR4 y AR5, una selección incorrecta del cluster hace que el sistema falle, sin embargo, nuestro sistema es capaz de soportar ese tipo de errores. Los resultados de AR1 también demuestran que el estudio de la ambigüedad de la base de datos de manera off-line y la construcción de las *D-Sphere* reduce el costo computacional del sistema (vea AR1 vs AR5) y que la estrategia de calcular, en tiempo real, la próxima posición del sensor aumenta la tasa de reconocimiento (AR1 vs AR4)

4. Conclusiones

Este artículo ha presentado un framework para el desarrollo de sistemas activos de reconocimiento de objetos basados en visión monocular que no está basado en modelos probabilísticos, lo que permite emplear vectores de características precisos durante la estimación de la pose del objeto. Además, implementa un sistema de clasificación empleando SVM, lo que hace más robusto el sistema de reconocimiento en entornos no controlados. También se demuestra que el sistema es capaz de solucionar eficientemente los problemas de incertidumbre/ambigüedad inherentes a bases de datos que contienen gran cantidad de objetos de forma libre. La segmentación de la base de datos en clusters y la estrategia de convertir los clusters en *D-Spheres*, permite realizar tareas de reconocimiento de objetos con una excelente relación entre la tasa de reconocimiento y la eficiencia computacional. La viabilidad y efectividad de la estrategia propuesta fueron experimentalmente verificadas en una plataforma

de trabajo real y comparadas con otros sistemas de reconocimiento 3D, demostrando ser más robusta y eficiente computacionalmente.

English Summary

Heuristic Framework to Develop Active Object Recognition

Abstract

This paper presents a framework for the development of active systems for object recognition. The proposed framework addresses the problem of uncertainty in object recognition based on monocular vision using a heuristic model that allow the implementation of the shape recognition stage by means of feature vectors without stochactic properties (PCA). The strategy employed to develop the proposed active recognition system is based on grouping the views of the objects in the database and clusters. From the study of the information contained in the cluster are efficiently developed the classification task, selection of sensor positions and calculation of the evidence. The classification algorithm uses a support vector machine (SVM) model to provide robustness to small deformations in the appearance of objects by noise, lighting changes, variations in the point of view. The sensor planing stage, which aims to reduce uncertainty using a minimum number of sensor movements, is based on the *D-Sphere* model. Each cluster is represented by a *D-Sphere* which allows, in an off-line process, to evaluate the uncertainty between objects hypothesis. This method has been tested in a real environment with a robot equipped with a webcam on the end-effector.

Keywords:

Free-form object recognition, active recognition system, SVM, clustering, next bext view

Agradecimientos

Este trabajo ha sido subvencionado por el Programa de Investigación del Gobierno Español con Referencia DPI2009-09956 (MICINN) y por el Fondo Social Europeo.

Referencias

Abdi, H., Williams, L. J., 2010. Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics 2 (4), 433–459.

Borotschnig, H., Paletta, L., Pinz, A., 1999. A comparison of probabilistic, possibilistic and evidence theoretic fusion schemes for active object recognition. Computing 62 (4), 293–319.

Bramao, I., Reis, A., Petersson, K. M., Faísca, L., 2011. The role of color information on object recognition: A review and meta-analysis. Acta Psychologica 138 (1), 244 – 253.

Campbell, R. J., Flynn, P. J., 2001. A survey of free-form object representation and recognition techniques. Computer Vision and Image Understanding 81, 166–210.

Cristianini, N., Shawe-Taylor, J., 2010. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press.

Deinzer, F., Denzler, J., Derichs, C., Niemann, H., 2006. Integrated Viewpoint Fusion and Viewpoint Selection for Optimal Object Recognition. In: British Machine Vision Conference 2006. Vol. 1. pp. 287–296.

Deinzer, F., Denzler, J., Niemann, H., 2003. Viewpoint Selection - Planning Optimal Sequences of Views for Object Recognition. In: Petkov, N., Westenberg, M. A. (Eds.), Computer Analysis of Images and Patterns - CAIP '03. Lecture Notes in Computer Science. -, pp. 65–73.

Flusser, J., 2000. On the independence of rotation moment invariants. Pattern Recognition 33 (9), 1405 – 1410.

González, E., Adán, A., Batlle, V. F., 2012. 2d shape representation and similarity measurement for 3d recognition problems: An experimental analysis. Pattern Recognition Letters 33 (2), 199–217.

González, E., Adán, A., Feliú, V., 2004. Descriptores de fourier para la identificación y posicionamiento de objetos en entornos 3d. XXV Jornadas de automática.

González, E., Adán, A., Feliú, V., Sánchez, L., June 2008a. Active object recognition based on fourier descriptors clustering. Pattern Recogn. Lett. 29, 1060–1071.

González, E., Adán, A., Feliú, V., Sánchez, L., 2008b. A solution to the next best view problem based on d-spheres for 3d object recognition. In: Proceedings of the Tenth IASTED International Conference on Computer Graphics and Imaging. CGIM '08. ACTA Press, pp. 286–291.

Gremban, K. D., Ikeuchi, K., 1994. Planning multiple observations for object recognition. International Journal of Computer Vision 12 (2-3), 137–172.

Hu, M.-K., february 1962. Visual pattern recognition by moment invariants. Information Theory, IRE Transactions on 8 (2), 179 –187.

Janoos, F., 2006. A perceptual study of shape metrics.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., Wu, A. Y., July 2002. An efficient k-means clustering algorithm: Analysis and implementation. IEEE Trans. Pattern Anal. Mach. Intell. 24, 881–892.

Kopp-Borotschnig, H., Paletta, L., Prantl, M., Pinz, A., 2000. Appearance-based active object recognition. Image Vision Comput. 18 (9), 715–727.

Kovacic, S., Leonardis, A., Pernus, F., 1998. Planning sequences of views for 3-d object recognition and pose determination. Pattern Recognition 31 (10), 1407–1417.

Kragic, D., Christensen, H. I., 2002. Survey on visual servoing for manipulation. Tech. rep., Computational Vision and Active Perception Laboratory.

Liu, C.-H., Tsai, W.-H., May 1990. 3d curved object recognition from multiple 2d camera views. Comput. Vision Graph. Image Process. 50, 177–187.

Rai, P., Singh, S., October 2010. Article:a survey of clustering techniques. International Journal of Computer Applications 7 (12), 1–5, published By Foundation of Computer Science.

Roy, S. D., Chaudhury, S., Banerjee, S., 2004. Active recognition through next view planning: a survey. Pattern Recognition 37 (3), 429–446.

S.A., H., A.C, K., 1999. Strategies Using Dempster-Shafer Belief Accumulation.

Sipe, M. A., Casasent, D., 2002. Feature space trajectory methods for active computer vision. IEEE Trans. Pattern Anal. Mach. Intell. 24 (12), 1634–1643.

Steinwart, I., Christmann, A., 2008. Support Vector Machines, 1st Edition. Springer Publishing Company, Incorporated.

UCLM, 2011. 3dsl: Dataset object models.
URL: `http://isa.esi.uclm.es/descarga-objetos-adan/`

Yuan, D., Street, W. N., 2007. Hacs: Heuristic algorithm for clustering subsets. In: SDM.

Zahn, C. T., Roskies, R. Z., March 1972. Fourier descriptors for plane closed curves. IEEE Trans. Comput. 21, 269–281.

Zitová, B., Flusser, J., 2003. Image registration methods: a survey. Image Vision Comput. 21 (11), 977–1000.